LA-5059-MS

INFORMAL REPORT

A. 2358

11-28-72
215

Specifications for CAMAC Subroutines

los alamos
scientific laboratory
of the University of California
LOS ALAMOS, NEW MEXICO 87544

MASTER

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Atomic Energy Commission, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

In the interest of prompt distribution, this LAMS report was not edited by the Technical Information staff.

Printed in the United States of America. Available from
National Technical Information Service
U. S. Department of Commerce
5285 Port Royal Road
Springfield, Virginia 22151
Price: Printed Copy \$3.00; Microfiche \$0.95

LA-5059-MS
Informal Report
UC-28 and 32

ISSUED: October 1972

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Atomic Energy Commission, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

Specifications for CAMAC Subroutines

by

Richard F. Thomas, Jr.

This work funded by the U. S. Atomic Energy Commission's Division of Research.

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

GG

SPECIFICATIONS FOR CAMAC SUBROUTINES

Richard F. Thomas, Jr.

ABSTRACT

This report has been prepared under the guidance and sponsorship of the CAMAC Software Working Group of the U. S. AEC NIM Committee. It contains descriptions of six FORTRAN-compatible subroutines recommended for use with CAMAC systems. These subroutines permit the most general operations on a CAMAC system to be specified in a high-level programming language. Two of the subroutines implement the fundamental CAMAC functions; three of them implement the three uses of the Q response defined in TID-25875; the remaining subroutine provides a flexible alternative for CAMAC addressing.

The NIM-CAMAC Software Working Group has agreed that the specification of a standard set of subroutines for performing CAMAC operations would be of use in facilitating communication, permitting interchange of programs and saving of design effort in the implementation of CAMAC systems. The subroutines should be applicable to as many languages as possible, but must be compatible with FORTRAN, which is the most widely used high-level language in the U. S. at this time. This report describes six subroutines which the Working Group recommends for standard usage. Other subroutines are under active consideration, but it is felt that the best interests of the CAMAC user community are served by rapid publication of the specifications which have been agreed upon to date.

The approach which has been followed in the following specifications is that a single subroutine, "CMCBSC," capable of performing a very general CAMAC operation, is specified in terms of its inputs and outputs. Then other, more specialized, subroutines are defined as FORTRAN subroutines which make use of the fundamental subroutine, "CMCBSC." It should be especially noted that although all subroutines except "CMCBSC" are defined as FORTRAN subroutines, an individual installation will quite likely find it advantageous to implement many of them in assembly language, or perhaps in other special ways, in

order to take advantage of local hardware and speed up their execution. In these cases, the FORTRAN subroutine will serve only as a definition. However, any installation which can provide a local version of "CMCBSC" and a FORTRAN compiler will have immediately available a library of CAMAC procedures.

The following general remarks apply to the descriptions of all the subroutines which follow. In specifying subroutine arguments names of arguments which may be or must be arrays end with the letter "A," and all arguments are integer variables in the FORTRAN sense. All subroutines, function and named common names, i.e., all global names beginning with the letters "CMC" are reserved for possible future extensions of the standard subroutines. To the maximum extent practical in the specifications argument names and meanings are identical from one subroutine to another. Certain arguments (especially crate number, repeat count, data block and error vector) are explained in detail in the description of subroutine "CMCBSC" and only briefly in the definitions of subsequent subroutines. However, the full definition is intended to apply in every case.

It is necessary for many of the subroutines to have access to the values of certain constants which describe the system, in particular the word length of the computer, the number of modules per crate, and the number of crates per branch. In order that the

subroutines can be written in a general way and be reasonably independent of the actual values of these quantities, a named common which contains the values for the system is defined, a sample "BLOCK DATA" subroutine is given in Fig. 1 for a 16-bit computer with normal values for the quantities.

```

BLOCK DATA
COMMON/CMCCOM/NCOMP,NOMOD,NOCRT

C
C   STANDARD NAMED COMMON FOR CAMAC SUBROUTINES.
C
C   NCOMP: NO. OF COMPUTER-LENGTH WORDS NECESSARY TO CONTAIN ONE
C           CAMAC-LENGTH WORD.
C   NOMOD: NO. OF MODULES (MAXIMUM) IN A CRATE.
C   NOCRT: NO. OF CRATES PER BRANCH.
C
DATA NCOMP, NOMOD,NOCRT/2,23,7/
END

```

Fig. 1

I. CMCBSC

Purpose

To perform a single CAMAC function at a single CAMAC address one or more times.

Calling Sequence

CALL CMCBSC(F,B,C,N,AD,LN,DATA,Q,ERRORA)

Arguments

F: CAMAC function code.

B: Branch number.

C: Crate number.

The crate number may be positive, negative, or zero. If it is positive and less than or equal to the number of crates per branch, a single crate operation is specified. Crate number = 0 indicates a branch controller function. If the crate number is between -1 and -127, then its absolute value, taken as a binary number, indicates which crates are to participate in a parallel operation. If the nth bit from the right is a "1," then crate n is addressed (e.g., C = -3 addresses crates 1 and 2). Crate number = -128 indicates all on-line crates are addressed in parallel.

N: Station number.

AD: Subaddress.

LN: Repeat count and word length specification.

If F specifies a data transfer function (read or write), the absolute value of LN indicates the number of words to transmit;

the sign indicates computer-length words if positive and CAMAC-length words if negative. If F specifies a control or test function, LN must be positive, and it indicates the number of times to repeat the operation.

DATA: Data block.

"DATA" must be an array of sufficient length to supply or to receive the data transmitted by the subroutine while executing the CAMAC command. "DATA" supplies data during write operations and receives data during read operations. If the computer word length is less than 24 bits and LN is negative for a read or write function, each CAMAC word will be stored in a contiguous block within the array "DATA" which will be the smallest integral number of words required to hold 24 bits. The word within the block having the greatest address is filled with the lowest order bits of the CAMAC word. Computer words having successively smaller addresses are filled with successively higher order bits of the CAMAC word until the last, which is filled out with high-order zeros if necessary. Note that although B,C,N, and AD remain constant in this subroutine, the computer storage address is incremented for each transmission.

Q: Q response from the last execution of the function.

ERRORA: Error vector.

Errors detected by the hardware or errors detected in arguments are indicated in this array. If the first word of the error vector is set to 0, i.e., ERRORA(1)=0, no errors were detected. If ERRORA(1)>0, then it

contains an error code, and the following three words may contain supplementary information about the error. The codes are:

1. Dataway timing error
2. Branch highway timing error (e.g., offline crate)
3. Illegal branch number
4. Illegal crate number (e.g., crate number greater than 7)
5. No X response
6. Illegal function code (e.g., not in the range 0 to 31)
7. Illegal station number (e.g., not in the range 0 to 31)
8. Illegal subaddress (e.g., not in the range 0 to 15)
9. Illegal value of LN
10. Starting CAMAC address is less than ending CAMAC address (see CMCASC)

ERRORA(2) contains the execution count at the time the error was detected. The subroutine stops execution when an error occurs and does not attempt to complete the

requested number of function executions. ERRORA(3) is used as a cumulative error count. Whenever an error is detected, its value is increased by 1.

II. CMCSEQ

Purpose

To execute a single CAMAC function at a succession of CAMAC addresses.

Calling Sequence

CALL CMCSEQ(F,BA,CA,NA,ADA,LN,DATA,Q,ERRORA)

Arguments

F: CAMAC function code.

BA: Branch numbers.

CA: Crate numbers.

NA: Station numbers.

ADA: Subaddresses.

LN: Repeat count and word length specification.
(+ = computer length; - = CAMAC length.)

DATA: Data block.

Q: Q response from last execution of function.

ERRORA: Error vector. (Integer array of length 4)

FORTTRAN Definition (See Fig. 2.)

```

SUBROUTINE CMCSEQ(F,BA,CA,NA,ADA,LN,DATA,Q,ERRORA)
COMMON/CMCCOM/NCOMP,NOMOD,NOCRT
INTEGER F,BA(2),CA(2),NA(2),ADA(2),DATA(2),Q,ERRORA(4)
C
C   SET NUMBER OF WORDS PER TRANSFER.
C
  NWDS=NCOMP
  IF(LN.GT.0)NWDS=1
C
C   SET NUMBER OF EXECUTIONS.
C
  K=IABS(LN)
  IF(K.EQ.0) GO TO 30
C
C   LOOP THROUGH CALL TO SUBROUTINE CMCBSC.
C
  DO 10 I=1,K
    M=I*NWDS-NWDS+1
    CALL CMCBSC(F,BA(I),CA(I),NA(I),ADA(I),ISIGN(1,LN),DATA(M),Q,
1    ERRORA)
C
C   CHECK FOR ERROR.
C
  IF(ERRORA(1).NE.0) GO TO 20
10 CONTINUE
  RETURN
20 ERRORA(2)=I
  RETURN
30 ERRORA(1)=9
  ERRORA(2)=0
  ERRORA(3)=ERRORA(3)+1
  RETURN
END
```

Note: Subroutine "CMCSEQ" could just as well serve as the fundamental CAMAC subroutine as subroutine "CMCBSC" since for LN=1 or LN=-1 they are identical in function, and any other sequence of CAMAC operations can be constructed from them.

Fig. 2

III. CMCASC

Purpose

To execute a specified CAMAC function in the address scan mode.

Calling Sequence

CALL CMCASC(F,BI,CI,NI,AI,BF,CF,NF,AF,LN,DATA,ERRORA,NEX)

Arguments

F: CAMAC function code.

BI: Initial branch number.

CI: Initial crate number.

NI: Initial station number.

AI: Initial subaddress.

BF: Final branch number.

CF: Final crate number.

NF: Final station number.

AF: Final subaddress.

LN: Execution limit and word size indication.

(+ = computer length; - = CAMAC length.)

DATA: Dimensioned array for data sent or received.

ERRORA: Error vector.

NEX: Number of times function was executed with a Q=1 response, (number of words transmitted if function is read or write).

Addressing Rule

The specified function is executed first at the address given by BI,CI,NI,AI. Then if the Q response is 1, the subaddress is incremented by 1 and the index into the data block incremented by either 1 or the number of computer words necessary to contain a CAMAC word, depending on the sign of LN, and the function executed at this new subaddress. If the Q response is 0, the subaddress is set to zero, the data block index is not changed, and the station number is incremented by 1 for the next execution. When the station number is incremented beyond the value of "NOMOD," it is set back to 1 and the crate number is incremented. If the crate number is incremented beyond the value of "NOCRT," it is set back to 1, and the branch number is incremented. The total number of executions of the function is limited by the magnitude of LN and by the stated final values of the CAMAC address, BF,CF,NF,AF. Special attention must be paid to the X response in the address scan mode. When a module responds with Q=0, it may, at the same time, give X=1 or X=0; consequently, in this mode of access, the X response is ignored when the Q response is 0.

FORTTRAN Definition (See Fig. 3.)

IV. CMCRTPT

Purpose

To execute a specified CAMAC function in the repeat mode.

Calling Sequence

CALL CMCRTPT(F,B,C,N,AD,LN,DATA,ERRORA)

Arguments

F: CAMAC function code.

B: Branch number.

C: Crate number.

N: Station number.

Ad: Subaddress.

LN: Execution limit and word size indication.

(+ = computer length; - = CAMAC length.)

DATA: Dimensions array for data sent or received.

ERRORA: Error vector.

Execution Rule

In the repeat mode, the CAMAC address (B,C,N,A) is never changed, but the single address is expected to supply many words of data. Q is used as a timing signal; Q=1 indicates that the previously executed function succeeded; Q=0 indicates that the module was not ready to accept the function and that the controller should try again.

FORTTRAN Definition (See Fig. 4.)

V. CMCSTP

Purpose

To execute a specified CAMAC function in the stop mode.

Calling Sequence

CALL CMCSTP(F,B,C,N,AD,LN,DATA,ERRORA,NEX)

Arguments

F: CAMAC function code.

B: Branch number.

C: Crate number.

N: Station number.

Ad: Subaddress.

LN: Execution limit and word size indication.

(+ = computer length; - = CAMAC length.)

DATA: Dimensioned array for data sent or received.

ERRORA: Error vector.

NEX: Number of times function was executed.

Execution Rule

In the stop mode, the CAMAC address (B,C,N,A) is never changed, but the single address is expected to supply (or accept) many words of data. It is

```

SUBROUTINE CMCASC(F,BI,CI,NI,AI,BF,CF,NF,AF,LN,DATA,ERRORA,NEX)
COMMON/CMCCOM/NCOMP,NOMOD,NOCRT
INTEGER F,BI,CI,AI,BF,CF,AF,DATA(2),ERRORA(4)
INTEGER B,C,A,Q

C
C   SET MAXIMUM NUMBER OF EXECUTIONS.
C
C   K=IABS(LN)

C
C   SET NUMBER OF WORDS PER DATA TRANSMISSION.
C
C   NWDS=NCOMP
C   IF(LN.GT.0) NWDS=1

C
C   CHECK FOR ARGUMENT ERROR.
C
C   IF(K.EQ.0) GO TO 50
C   IF(BF.GT.BI) GO TO 5
C   IF(BF.LT.BI) GO TO 60
C   IF(CF.GT.CI) GO TO 5
C   IF(CF.LT.CI) GO TO 60
C   IF(NF.GT.NI) GO TO 5
C   IF(NF.LT.NI) GO TO 60
C   IF(AF.LT.NI) GO TO 60

C
C   SET INITIAL VALUES OF B, C, N, A.
C
C   5 B=BI
C   C=CI
C   N=NI
C   A=AI

C
C   J IS USED TO INDEX THE ARRAY "DATA".
C   I COUNTS THE NUMBER OF EXECUTIONS OF THE FUNCTION.
C
C   J=1
C   I=0

C
C   HAVE WE EXCEEDED THE MAXIMUM ALLOWABLE NUMBER OF EXECUTIONS OF
C   THIS FUNCTION?
C
C   10 IF(I.GE.K) GO TO 30

C
C   EXECUTE THE FUNCTION ONCE.
C
C   CALL CMCBSC(F,B,C,N,A,ISIGN(1,LN),DATA(J),Q,ERRORA)

C
C   CHECK FOR ERROR.
C
C   IF(ERRORA(1).NE.0) GO TO 20

C
C   CHECK Q.
C
C   IF(Q.EQ.0) GO TO 40

C
C   INCREMENT THE CORE ADDRESS INDEX AND THE CAMAC SUBADDRESS.
C
C   I=I+1
C   J=J+NWDS
C   A=A+1
C   IF(A.GT.15) GO TO 40

```



```

C
C      CHECK FOR END OF RANGE.
C
20  IF(B.LT.9F) GO TO 10
    IF(B.GT.9F) GO TO 30
    IF(C.LT.CF) GO TO 10
    IF(C.GT.CF) GO TO 30
    IF(N.LT.NF) GO TO 10
    IF(N.GT.NF) GO TO 30
    IF(A.LE.AF) GO TO 10
    GO TO 30
C
C      B,C,N,A, OR I HAS EXCEEDED LIMIT; HALT EXECUTION OF
C      CAMAC FUNCTION.
C
28  IF((ERRORA(1).EQ.5).AND.(D.EQ.0))GO TO 38
30  NEX=I
    RETURN
C
C      GO TO NEXT MODULE
C
38  ERRORA(1)=0
    ERRORA(3)=ERRORA(3)-1
40  A=0
    N=N+1
    IF(N.LE.NOMOD) GO TO 20
C
C      GO TO NEXT CRATE.
C
    N=1
    C=C+1
    IF(C.LE.NOCRT) GO TO 20
C
C      GO TO NEXT BRANCH.
C
    C=1
    B=B+1
    GO TO 20
C
C      ERRORS DETECTED IN ARGUMENTS.
C      1. LN=0
C
50  ERRORA(1)=9
    ERRORA(2)=0
    ERRORA(3)=ERRORA(3)+1
    RETURN
C
C      2. (BI,CI,NI,AI) IS GREATER THAN(BF,CF,NF,AF)
C
60  ERRORA(1)=10
    RETURN
    END

```

Fig. 3
(Page 2 of 2)

```

SUBROUTINE CMCRPT(F,B,C,N,AD,LN,DATA,ERRORA)
COMMON/CMCCOM/NCOMP,NOMOD,NOCRT
INTEGER F,B,C,AD,DATA(2),ERRORA(4),Q
C
C   SET NO. OF COMPUTER WORDS PER TRANSFER.
C
NWDS=NCOMP
IF(LN.GT.0) NWDS=1
C
C   SET NUMBER OF EXECUTIONS.
C
K=IABS(LN)
C
C   CHECK FOR ZERO.
C
IF(LN.EQ.0) GO TO 40
C
C   SET EXECUTION PARAMETER FOR "CALL CMCBSC"
C
L=[SIGN(1,LN)
C
C   INITIALIZE LOOP.
C
J=1
I=0
10 IF(I.GE.K) RETURN
C
C   INCREMENT TALLY
C
I=I+1
C
C   ATTEMPT TO EXECUTE FUNCTION.
C
20 CALL CMCBSC(F,B,C,N,AD,L,DATA(J),Q,ERRORA)
IF(ERRORA(1).NE.0) GO TO 30
IF(Q.EQ.0) GO TO 20
C
C   INCREMENT STORAGE ADDRESS AND LOOP BACK.
C
J=J+NWDS
GO TO 10
C
C   ERROR. STORE EXECUTION COUNT IN ERRORA(2).
C
30 ERRORA(2)=I
RETURN
C
C   LN=0. RETURN ERROR 9.
C
40 ERRORA(1)=9
ERRORA(2)=0
ERRORA(3)=ERRORA(3)+1
RETURN
END

```

Fig. 4

assumed able to supply or accept a word of data whenever the controller addresses it, until the block is exhausted. The controller may terminate the process if the number of executions exceeds the limit given by the magnitude of LN, or the module may terminate the process by responding with Q=0. Q=0 implies that no data was sent or accepted during the CAMAC cycle for which it is the response. Q=1 indicates normal execution of the function.

FORTRAN Definition (See Fig. 5.)

VI. CMCLUP

Purpose

To execute a specified CAMAC function at a hierarchical sequence of addresses with the option of skipping certain portions of the sequence based on the Q response.

Calling Sequence

CALL CMCLUP(F,BA,LB,CA,LC,NA,LKN,ADA,LAD,LN,DATA,Q,QON,ERRORA,NEX)

```

SUBROUTINE CMCLUP(F,B,C,N,AD,LN,DATA,ERRORA,NEX)
COMMON/CMCCOM/NCOMP,NOMOD,NOCRT
INTEGER F,B,C,AD,DATA(2),ERRORA(4),Q
C
C   SET NO. OF COMPUTER WORDS PER TRANSFER
C
  NWDS=NCOMP
  IF(LN.GT.0) NWDS=1
C
C   SET NO OF EXECUTIONS.
C
  K=1ABS(LN)
  IF(K.EQ.0)GO TO 30
C
C   SET EXECUTION PARAMETER FOR "CALL CMCBSC"
C
  L=ISIGN(1,LN)
C
C   INITIALIZE LOOP.
C
  J=1
  I=0
C
C   TEST FOR END OF EXECUTION.
C
  10 IF(I.GE.K) RETURN
C
C   INCREMENT EXECUTION TALLY.
C
  I=I+1
C
C   ATTEMPT TO EXECUTE FUNCTION.
C
  CALL CMCBSC(F,B,C,N,AD,L,DATA(J),Q,ERRORA)
  IF(ERRORA(1).NE.0) GO TO 20
  IF(Q.EQ.0) GO TO 20
  J=J+NWDS
  GO TO 10
C
C   ERROR. SET EXECUTION COUNT IN NEX AND EXIT.
C
  20 NEX=I
  RETURN
C
C   ERROR: LN = 0.
C
  30 ERRORA(1)=9
  ERRORA(2)=0
  ERRORA(3)=ERRORA(3)+1
  GO TO 20
END

```

Fig. 5

Arguments

F: CAMAC function code.
BA: Branch number array.
LB: Number of branch numbers.
CA: Crate number array.
LC: Number of crate numbers.
NA: Module number array.
LKN: Number of module numbers.
ADA: Subaddress array.
LAD: Number of subaddresses.
LN: Execution limit and word size indication.
(+ = computer length; - = CAMAC length.)
DATA: Data array.
Q: Q response of last execution.
QON: Control argument indicating whether or not
to use Q response for skipping addresses.
1 means use Q; Q means ignore Q.
ERRORA: Error vector.
NEX: Number of times the function was executed,
or the number of times the function was
executed with a Q=1 response if QON=1,
(number of words transmitted for read or
write functions).

Execution Rule

"BA," "CA," "NA," and "ADA," are each one-dimensional arrays containing lists of branch, crate, module, and subaddress numbers respectively. The program executes, or attempts to execute, the function at each CAMAC address which can be formed by combining the elements of these arrays. It begins by choosing the first element of each array and advances the address by choosing the next element of the subaddress array, etc., scanning the subaddresses most frequently, the module numbers next, etc., until the last element of each array is used. If QON=1, a Q=0 response will cause the module index to advance and the subaddress index to be set to 1.

FORTRAN definition (See Fig. 6.)

VII. REFERENCES

1. "CAMAC: A Modular Instrumentation System for Data Handling," TID-25875.
2. "CAMAC: Organization of Multi-Crate System," TID-25876.
3. "Supplement to CAMAC Instrumentation System Specifications," TID-25877.

```
      SUBROUTINE CMCLUP(F,BA,LB,CA,LC,NA,LKN,ADA,LAD,LN,DATA,Q,QON,  
1      ERRORA,NEX)  
      COMMON/CMCCOM/NCOMP,NOMOD,NOCRT  
      INTEGER F,BA(LB),CA(LC),NA(LKN),ADA(LAD),DATA(2),Q,QON,ERRORA(4)  
  
      C  
      C      SET NUMBER OF COMPUTER WORDS PER TRANSFER.  
      C  
      NWDS=NCOMP  
      IF(LN.GT.0) NWDS=1  
  
      C  
      C      SET EXECUTION LIMIT.  
      C  
      K=IABS(LN)  
  
      C  
      C      CHECK LIMIT FOR ZERO.  
      C  
      IF(K.EQ.0) GO TO 60  
  
      C  
      C      SET EXECUTION PARAMETER FOR "CALL CMCBSC".  
      C  
      L=ISIGN(1,LN)  
  
      C  
      C      INITIALIZE INDEXES FOR EXECUTION COUNT AND DATA TRANSFER. I  
      C      COUNTS EXECUTION OF F; J INDEXES "DATA".  
      C  
      I=0  
      J=1
```

Fig. 6
(Page 1 of 2)

```

C
C   SET UP FOUR NESTED DO'S TO EXECUTE THE ADDRESSING ALGORITHM.
C
DO 40 MB=1,LB
DO 40 MC=1,LC
DO 40 MN=1,LKN
DO 30 MAD=1,LAD

C
C   ATTEMPT TO EXECUTE FUNCTION ONE TIME.
C
CALL CMCBSC(F,BA(MB),CA(MC),NA(MN),ADA(MAD),L,DATA(J),Q,ERRORA)

C
C   IS Q BEING CHECKED?
C
IF(QON.EQ.0) GO TO 20

C
C   YES: IF Q=1,PROCEED NORMALLY. IF Q=0, IGNORE X RESPONSE AND
C   BRANCH TO END OF N LOOP.
C
IF(Q.EQ.1) GO TO 20
IF((ERRORA(1).EQ.5).OR.(ERRORA(1).EQ.0)) GO TO 40
GO TO 70

C
C   INCREMENT DATA INDEX AND EXECUTION COUNT. TEST EXECUTION LIMIT.
C
20 I=I+1
J=J+NWDS
IF(I.GE.K) GO TO 50

C
C   DO NORMAL ERROR CHECK.
C
IF(ERRORA(1).NE.0) GO TO 70

C
C   END OF SUBADDRESS LOOP.

C
30 CONTINUE

C
C   END OF BRANCH, CRATE, MODULE LOOPS.
C
40 CONTINUE

C
C   SET EXECUTION COUNT AND RETURN
C
50 NEX=I
RETURN

C
C   LN=0. GIVE TYPE 9 ERROR.
C
60 ERRORA(1)=9
ERRORA(2)=0
ERRORA(3)=ERRORA(3)+1
RETURN

C
C   ERROR. SET ERRORA(2) TO EXECUTION COUNT.
C
70 ERRORA(2)=I
GO TO 50
END

```

Fig. 6
(Page 2 of 2)